SYSTEMS AND METHODS FOR ASSOCIATING A KEYWORD WITH A USER INTERFACE AREA

RELATED APPLICATIONS

[0001]   This application relates to:

Attorney Docket No. GP-175-12-US, filed herewith, titled "Systems and Methods for Generating Multiple Implicit Search Queries";

Attorney Docket No. GP-175-13-US, filed herewith, titled "Systems and Methods for Extracting a Keyword from an Event";

Attorney Docket No. GP-175-14-US, filed herewith, titled "Systems and Methods for Weighting a Search Query Result";

Attorney Docket No. GP-175-15-US, filed herewith, titled "Systems and Methods for Refreshing a Content Display";

Attorney Docket No. GP-175-16-US, filed herewith, titled "Systems and Methods for Constructing and Using a User Profile";

Attorney Docket No. GP-175-17-US, filed herewith, titled "Systems and Methods for Identifying a Named Entity";

Attorney Docket No. GP-175-18-US, filed herewith, titled "Systems and Methods for Analyzing Boilerplate";

Attorney Docket No. GP-175-39-US, filed herewith, titled "Systems and Methods for Ranking Implicit Search Results";

Attorney Docket No. GP-175-40-US, filed herewith, titled "Systems and Methods for Generating a User Interface"; and

Attorney Docket No. GP-175-51-US, filed herewith, titled "Systems and

Methods for Providing Search Results,"

the entirety of all of which are incorporated herein by reference.

FIELD OF THE INVENTION

[0002]   The present invention relates generally to methods and systems for

information retrieval. The present invention relates particularly to systems and methods

for associating a keyword with a user interface area.

BACKGROUND

[0003]   Conventional search engines receive a search query from a user and execute a

search against a global index. Such conventional search engines typically use one or

more conventional methods for performing a search. For example, one known method,

described in an article entitled "The Anatomy of a Large-Scale Hypertextual Search

Engine," by Sergey Brin and Lawrence Page, assigns a degree of importance to a

document, such as a web page, based on the link structure of the web. The search results

are often presented in a list format, including article identifiers and brief snippets about

the documents in a web page that can be resized.

[0004]   A user may also have access to other information stored on the user's local

machine or on other storage media accessible via a network that is relevant to a user.

Typically, a user enters an explicit search that includes keywords and that is executed

against a global or local index (As used herein, a "keyword" or "keywords" is defined

broadly to mean words, sequences of words, acronyms or other characters, including

spaces, based upon which a search may be performed).

SUMMARY

[0005]   Embodiments of the present invention provide systems and methods for

associating a keyword with a user interface area. In one embodiment of the present

invention, a search system, which may be implemented in hardware, software or a

combination thereof, associates a keyword with a first user interface area. The system

receives a signal that the first user interface area is inactive and that a second user

interface area is active. In response, the system generates an implicit search query that

includes the keyword. In one embodiment, a computer-readable medium (such as, for

example random access memory or a computer disk) comprises code from carrying out

such a method.

[0006]   These exemplary embodiments are mentioned not to limit or define the

invention, but to provide examples of embodiments of the invention to aid understanding

thereof. Exemplary embodiments are discussed in the Detailed Description, and further

description of the invention is provided there. Advantages offered by the various

embodiments of the present invention may be further understood by examining this

specification.

BRIEF DESCRIPTION OF THE FIGURES

[0007]   These and other features, aspects, and advantages of the present invention are

better understood when the following Detailed Description is read with reference to the

accompanying drawings, wherein:

Figure 1 is a block diagram illustrating an exemplary environment in which

one embodiment of the present invention may operate;

Figure 2 is a flowchart illustrating a method for associating a keyword with a user interface area in one embodiment of the present invention;

Figure 3 is a flowchart illustrating a method of reweighting a keyword based on an inactivity period in one embodiment of the present invention; and

Figure 4 is a block diagram of a user display in one embodiment of the present invention.

DETAILED DESCRIPTION

[0008] Embodiments of the present invention provide systems and methods for associating a keyword with a user interface area. Exemplary embodiments are described below.

System Architecture

[0009] Referring now to the drawings in which like numerals indicate like elements throughout the several figures, Figure 1 is a block diagram illustrating an exemplary environment for implementation of an embodiment of the present invention. While the environment shown reflects a client-side search engine architecture embodiment, other embodiments are possible.

[0010] The system 100 shown in Figure 1 includes multiple client devices 102a-n in communication with a server device 150 over a wired or wireless network 106. The network 106 shown comprises the Internet. In other embodiments, other networks, such as an intranet, may be used instead. Moreover, methods according to the present invention may operate within a single client device.

[0011]   The client devices 102a-n shown each includes a computer-readable medium 108. The embodiment shown includes a random access memory (RAM) 108 coupled to a processor 110. The processor 110 executes computer-executable program instructions stored in memory 108. Such processors may include a microprocessor, an ASIC, a state machine, or other processor, and can be any of a number of computer processors, such as processors from Intel Corporation of Santa Clara, California and Motorola Corporation of Schaumburg, Illinois. Such processors include, or may be in communication with, media, for example computer-readable media, which stores instructions that, when executed by the processor, cause the processor to perform the steps described herein.

[0012]   Embodiments of computer-readable media include, but are not limited to, an electronic, optical, magnetic, or other storage or transmission device capable of providing a processor, such as the processor 110 of client 102a, with computer-readable instructions. Other examples of suitable media include, but are not limited to, a floppy disk, CD-ROM, DVD, magnetic disk, memory chip, ROM, RAM, an ASIC, a configured processor, all optical media, all magnetic tape or other magnetic media, or any other medium from which a computer processor can read instructions. Also, various other forms of computer-readable media may transmit or carry instructions to a computer, including a router, private or public network, or other transmission device or channel, both wired and wireless. The instructions may comprise code from any suitable computer-programming language, including, for example, C, C++, C#, Visual Basic, Java, Python, Perl, and JavaScript.

[0013]   Client devices 102a-n can be connected to a network 106 as shown, or can be stand-alone machines. Client devices 102a-n may also include a number of external or internal devices such as a mouse, a CD-ROM, DVD, a keyboard, a display, or other input or output devices. Examples of client devices 102a-n are personal computers, digital assistants, personal digital assistants, cellular phones, mobile phones, smart phones, pagers, digital tablets, laptop computers, Internet appliances, and other processor-based devices. In general, the client devices 102a-n may be any type of processor-based platform that operates on any operating system, such as Microsoft® Windows® or Linux, capable of supporting one or more client application programs. For example, the client device 102a shown comprises a personal computer executing client application programs, also known as client applications 120. The client applications 120 can be contained in memory 108 and can include, for example, a word processing application, a spreadsheet application, an email application, an instant messenger application, a presentation application, an Internet browser application, a calendar/organizer application, and any other application or computer program capable of being executed by a client device.

[0014]   The user 112a can interact with the various client applications 120 and articles associated with the client applications 120 via various input and output devices of the client device 102a. Articles include, for example, word processor, spreadsheet, presentation, email, instant messenger, database, and other client application program content files or groups of files, web pages of various formats, such as HTML, XML, XHTML, Portable Document Format (PDF) files, and audio files, video files, or any other documents or groups of documents or information of any type whatsoever.

[0015]   The memory 108 of the client device 102a shown also contains a capture

processor 124, a queue 126, and a search engine 122. The client device 102a shown also

contains or is in communication with a data store 140. The search engine 122 can receive

an explicit query from the user 112a or generate an implicit query and retrieve

information from the data store 140 in response to the query.

[0016]   The search engine 122 shown contains an indexer 130, a query system 132,

and a formatter 134. Events, real-time and historical, contextual and indexable, and

performance data can be sent by the queue 126 to the query system 132 to provide the

query system 132 with information concerning current user context. The query system

132 can use this information to generate an implicit query. The query system 132 can

also receive and process explicit queries from the user 112a.

[0017]   The data store 140 can be any type of computer-readable media and can be

integrated with the client device 102a, such as a hard drive, or external to the client

device 102a, such as an external hard drive or on another data storage device accessed

through the network 106. The data store 140 may include any one or combination of

methods for storing data, including without limitation, arrays, hash tables, lists, and pairs.

[0018]   The data store 140 comprises a local index. The local index in the

embodiment shown in Figure 1 may comprise information, such as articles, which are

associated with the client device 102a, a user 112a of the client device 102a, or a group of

users of the client device 102a. For example, the local index in the data store 140 shown

in Figure 1 may comprise an index of articles created, edited, received, or stored by the

client user 112a using the client machine 102a, or articles otherwise associated with the client user 102a or the client machine 112a. The local index may be stored in a client machine, such as in data store 140, in a data store on a local network in a manner accessible by the client machine, on a server accessible to the client machine through the Internet, or in another accessible location.

[0019]   In contrast, a global index may comprise information relevant to many users or many servers, such as, for example, an index of web pages located on multiple servers in communication with the World Wide Web. One example of a global index is an index used by the Google(TM) search engine to provide search results in response to a search query.

[0020]   A single index may comprise both a local and a global index. For example, in one embodiment, an index may comprise both local and global information, and include a user or client identifier with the local information so that it may be identified with the user(s) or client(s) to which it pertains. Moreover, an index, local or global, may be present in one or multiple logical or physical locations.

[0021]   In the embodiment shown in Figure 1, a user 112a can input an explicit query into a search engine interface displayed on the client device 102a, which is received by the search engine 122. The search engine 122 can also generate an implicit query based on a current user context or state, which can be determined by the query system 132 from contextual real time events or other means. Based on the query, the query system 132 can locate relevant information in the data store 140 or other index and provide a result

set. In one embodiment, the result set comprises article identifiers identifying articles associated with the client applications 120 or client articles. Client articles stored in the data store 140 include articles associated with the user 112a or client device 102a, such as the word processing documents, previously viewed web pages and any other article associated with the client device 102a or user 112a. In another embodiment, the result set also comprises identifiers identifying articles located on the network 106 or network articles located by a search engine on a server device. Network articles include articles located on the network 106 not previously viewed or otherwise referenced by the user 112a, such as web pages not previously viewed by the user 112a.

[0022] The result sets comprise one or more article identifiers. An article identifier may be, for example, a Uniform Resource Locator (URL), a file name, a link, an icon, a path for a local file, or anything else that identifies an article. In the embodiment shown, an article identifier comprises a URL associated with an article.

[0023] Messaging articles stored in the data store 140 include user's emails, chat messages, and instant messaging messages. Each time a message is received, sent, modified, printed, or otherwise accessed, a record is stored in the data store 140. This information can later be searched to identify messages that should be displayed in the user interface.

[0024] An embodiment of the present invention may also store message threads in the data store 140. In such an embodiment, messages are related together by various attributes, including, for example, the sender, recipient, date/time sent and received, the

subject, the content, or any other attribute of the message. The related messages can then be retrieved as a thread, which may be treated as a document by the display processor 128.

[0025]   The formatter 134 can receive the search result set from the query system 132 of the search engine 122 and can format the results for output to a display processor 128. In one embodiment, the formatter 134 formats the results in XML or HTML. The display processor 128 can be contained in memory 108 and can control the display of the result set on a display device associated with the client device 102a. The display processor 128 may comprise various components. For example, in one embodiment, the display processor 128 comprises a Hypertext Transfer Protocol (HTTP) server that receives requests for information and responds by constructing and transmitting Hypertext Markup Language (HTML) pages. In one such embodiment, the HTTP server comprises a scaled-down version of the Apache Web server. In various embodiments, the functions described herein may be performed by various other components and devices.

[0026]   Through the client devices 102a-n, users 112a-n can communicate over the network 106, with each other and with other systems and devices coupled to the network 106. As shown in Figure 1, a server device 150 is also coupled to the network 106. In the embodiment shown, the search engine 122 can transmit a search query comprised of an explicit or implicit query or both to the server device 150. The user 112a can also enter a search query in a search engine interface, which can be transmitted to the server device 150. In another embodiment, the query signal may instead be sent to a proxy

server (not shown), which then transmits the query signal to server device 150. Other configurations are also possible.

[0027]   The server device 150 shown includes a server executing a search engine application program, such as the Google™ search engine. Similar to the client devices 102a-n, the server device 150 shown includes a processor 160 coupled to a computer-readable memory 162. Server device 150, depicted as a single computer system, may be implemented as a network of computer processors. Examples of a server device 150 are servers, mainframe computers, networked computers, a processor-based device, and similar types of systems and devices. The server processor 160 can be any of a number of or combination of computer processors, such as processors from Intel Corporation of Santa Clara, California and Motorola Corporation of Schaumburg, Illinois.

[0028]   Memory 162 contains the search engine application program, also known as a search engine 170. The search engine 170 locates relevant information in response to a search query from a client device 102a. The search engine 122 then provides the result set to the client device 102a via the network 106. The result set 134 comprises one or more article identifiers. An article identifier may be, for example, a URL, a file name, a link, an icon, a path for a local file, or anything else that identifies an article. In the embodiment shown, an article identifier comprises a URL associated with an article. The result set may include text, audio, video or any other type of content.

[0029]   In the embodiment shown, the server device 150, or related device, has previously performed a crawl of the network 106 to locate articles, such as web pages,

stored at other devices or systems connected to the network 106, and indexed the articles

in memory 162 or on another data storage device. In other embodiments, a crawl is not

performed. For example, in one embodiment, an index of articles is created manually.

[0030] It should be noted that embodiments of the present invention may comprise

systems having different architecture than that which is shown in Figure 1. For example,

in some systems according to the present invention, server device 104 may comprise a

single physical or logical server. The system 100 shown in Figure 1 is merely exemplary,

and is used to explain the exemplary methods shown in Figures 2 and 3.

Process

[0031] Various methods may be implemented in the environment shown in Figure 1

and other environments, according to the present invention. Methods according to the

present invention may be implemented by, for example, a processor-executable program

code stored on a computer-readable medium.

[0032] In one embodiment of the present invention, a system, such as indexer 130,

captures processor 124, or query system 132, associates a keyword with a first user

interface area. The query system 132 receives a signal that the first user interface area is

inactive and that a second user interface area is active. For example, in a Microsoft®

Windows operating environment, the query system 132 may intercept an application

programming interface (API) call directed to the operating system that instructs the

operating system to maximize a window or other user interface area in which an

application is executing. The query system 132 interprets this call as activating the

12

window that is the subject of the call and as inactivating all of the other windows in the user interface. The query system 132 may respond to receiving the call by executing a separate API call to determine all of the windows that are currently executing in the operating system and are inactive (i.e., all of the windows other than the active window).

[0033]   In response to receiving the signal that the first user interface area is inactive and the second is active, the query system 132 generates an implicit search query that includes the keyword. The keyword may be a single keyword or a plurality of keywords.

[0034]   In one embodiment, a computer program also identifies the keyword to be associated with the user interface area. For example, the user interface area may include a document, such as a word-processing document. In one embodiment, a computer program is able to retrieve or receive a keyword associated with the document.

[0035]   The query system 132 receives a second signal indicating that a second user interface area is active and generates an implicit query that includes the keyword from the first user interface area. In such an embodiment, use of the keyword in the implicit query may be discontinued after a period of time has elapsed, e.g., ten seconds in an exemplary embodiment of the present invention. In another such embodiment, the keyword or results associated with the keyword are downweighted after a period of time has elapsed. For example, after ten seconds the results associated with the keyword are downweighted by fifty percent; after 20 seconds, the results are downweighted by seventy-five percent; and after thirty seconds, use of the keyword is discontinued. In another embodiment, the

amount of downweighting is a function of the time since the keyword was extracted and/or the corresponding user interface area was active.

[0036]   The association between the keyword and the user interface area may persist. For example, the keyword and a user interface area identifier may be stored in memory. In one embodiment, the keyword is an attribute of an event as described in relation to Figure 1. The event may include other attributes, such as an identifier of the user interface area.

[0037]   In one embodiment of the present invention, the query system 132 submits the search query to a local or global index. In response, the query system 132 receives a result set and causes the result set to be output.

[0038]   In one embodiment, the query system 132 weights the keywords for the active user interface area (e.g., window) more heavily than keywords for inactive user interface areas. The weight may decrease proportionally to the time since the particular user interface area with which a keyword is associated was last active. If the user switches back to an inactive user interface area, then the keywords for that user interface area become weighted more heavily again. This is facilitated by keeping track of keywords for each active and inactive user interface area. In such an embodiment, sets of one or more keywords (e.g., from events), each of which has a corresponding user interface area identifier and a corresponding time. The weight at which the keywords are used varies depending on the active user interface area, the time since the keywords were created, and the time since the user interface area was last active if it is not currently active.

Older keywords have lower weight, and keywords associated with an inactive application have lower weight. If the user interface area becomes active again the weight increases. In another embodiment, the time since an application was last inactive is used in the weighting scheme, with lower weight going to keywords associated with user interface areas that have been inactive for a longer period.

[0039] In another embodiment, the system tracks the frequency and total amount of time that a user interface area is active. For example, if a user continually switches back and forth between an IM user interface area and several other applications, the system may recognize the repeated accesses to the IM user interface area and adjust the weight for keywords associated with that user interface area accordingly.

[0040] In one embodiment, adjustment of the weighting is performed using a step function that changes at certain predefined thresholds. In another embodiment, the adjustment is performed using a smooth function based on the age of the keyword.

[0041] Figure 2 is a flowchart illustrating a method for associating a keyword with a user interface area in one embodiment of the present invention. In the embodiment shown, a computer program, such as the query system 132, receives an event 202. The event signifies that some activity has occurred. For example, an event may signify that the user has received or sent an email, accessed a document, submitted an explicit query to a search engine, or performed some other activity. The event shown in Figure 2 signifies that some activity has occurred that is associated with a user interface area. For example, the user may have maximized, minimized, or restored a window, entered text

into a document, printed a document, selected text from a web page, or performed some other activity associated with the user interface area.

[0042]   The event includes attributes. For example, in the embodiment shown, the event includes an identifier of the user interface area containing the application that caused the event to be generated. For example, the user interface area may include a word-processing application. When the user completes a section of text, which is signaled when the user enters a punctuation mark, for example, an event is generated. The event includes an identifier of the word-processing application. The event may also include one or more keywords. The query system 132 associates the keyword(s) with the user interface area 204. The association may be stored in memory or otherwise persist. For instance, in one embodiment, a word processing application executes within a user interface area. When the user completes the typing of a sentence by, for example, entering a period, an event is generated. The event may comprise an identifier of the user interface area as well as a keyword or keywords from the sentence that the user just completed. The query system 132 saves the association between the keyword and the user interface area.

[0043]   Subsequently, the query system 132 receives a signal indicating that the user interface area with which the keyword or keywords is associated is active 206. For instance, the user may receive an email message and click on the email program to access the message. The query system 132 receives a signal indicating the user interface area associated with the email program is now active. Subsequently, the user clicks on the user interface area in which the word processing application is executing. The query

system 132 receives a signal indicating that the user interface area associated with the

email program is now inactive, and that the user interface area associated with the word

processing program is now active.

[0044]   After receiving the signal or in response to receiving the signal, the query

system 132 generates an implicit query 210. Since the user interface area is active, the

embodiment shown in Figure 2 includes the keywords associated with the user interface

area in the implicit query. If the user interface area is not currently active, the keywords

may or may not be included within the implicit query. In one embodiment, when

keywords associated with non-active user interface areas are used, they are associated

with lower weight than keywords corresponding to the active user interface area.

[0045]   The query system 132 causes the search result to be executed against a global

or local index 210. In the embodiment shown, the query system receives the search result

or results 212 and causes them to be output to a user 214. For example, the query system

132 may receive or generate an HTML page including the search results and transmit the

HTML page to the client 102a.

[0046]   Figure 3 is a flowchart illustrating a method of reweighting a keyword based

on an inactivity period in one embodiment of the present invention. In the embodiment

shown, the query system 132 receives an event 302. The query system 132 associates a

keyword of the event with an application user interface area 304.

[0047]   Subsequently, the query system 132 receives a signal indicating that the

application user interface area is inactive 306. For instance, if the user clicks on another

application in a different user interface area, the user interface area that is the source of

the signal becomes inactive. In the embodiment shown in Figure 3, once a user interface

area is inactive, implicit queries may rely relatively less on keywords associated with the

inactive user interface area than they would when the user interface area is active.

[0048]   The query system 132 next determines whether the amount of time that the

user interface area has been inactive exceeds a threshold 310. The threshold may be

predetermined or may be based on user or client-specific attributes. For example, the

user may explicitly set a limit for the amount of time keywords associated with an

inactive user interface area are utilized to perform implicit queries. In an embodiment of

the present invention, one or more thresholds may be used. In the embodiment shown, if

the inactive time period exceeds the threshold, the keyword is either excluded from

implicit queries or the results based on the keywords associated with the inactive user

interface area are downweighted in comparison to a keyword associated with an active

user interface area 312. In another embodiment, the amount of downweighting is a

function of the time since the keyword was extracted and/or the corresponding interface

area was active.

[0049]   The query system 132 executes the implicit query or causes the implicit query

to be executed on a global or local index 314. In response, the query system 132 receives

search results 316. The query system 132 then causes the query results to be output 318.

Example

[0050]   Figure 4 is a block diagram of a user display in one embodiment of the present

invention. The embodiment shown includes a user interface screen 402. The user

interface screen 402 includes a content display 404 for displaying the results of search

queries. The user interface screen also includes two user interface areas, application

windows 406 and 408. The application windows 406, 408 contain two applications,

Application 1 and Application 2, respectively, i.e., the two applications execute within

the respective windows. The user interface screen 402 also includes a toolbar 410.

[0051]   In a method according to one embodiment of the present invention, the user

enters data into the application executing within the first window 406. As the user enters

data, one or more keywords in the data are associated with that window and implicit

queries are generated based on the keywords. The results of the implicit queries are

displayed in the content display 404.

[0052]   The user then clicks on the window in which the second application is

executing 408. In the embodiment shown, implicit queries continue to be generated

periodically (e.g., once per second). Initially, after the first window becomes inactive,

the implicit queries include keywords from both the first window 406 and the second

window 408. After a specified period of time, the keywords associated with the inactive

window 406 are either downweighted in the query or result set, or the keywords are

excluded from implicit queries. The time period may be equal to zero, i.e., those

keywords are downweighted or excluded immediately when the window becomes

inactive. The association of the keyword and the window may persist even though the

keyword is not currently used in implicit queries.

[0053]   In one embodiment, when the user clicks on the window in which Application

1 is executing 406, the keywords that were previously associated with the window are

used to form implicit queries. In this way, the content display provides search results to

the user that are relevant to the application on which the user is focused or at least are

relevant to the currently active application.

General

[0054]   The foregoing description of embodiments of the invention has been presented

only for the purpose of illustration and description and is not intended to be exhaustive or

to limit the invention to the precise forms disclosed. Numerous modifications and

adaptations thereof will be apparent to those skilled in the art without departing from the

spirit and scope of the present invention.